

Characterizing Properties for Q-Clustering

Reza Bosagh Zadeh

*Institute for Computational and Mathematical Engineering
Stanford University*

REZAB@STANFORD.EDU

Gunnar Carlsson

*Department of Mathematics
Stanford University*

GUNNAR@MATH.STANFORD.EDU

Abstract

We uniquely characterize two members of the Q-Clustering family in an axiomatic framework. We introduce properties that use known tree constructions for the purpose of characterization. To characterize the Max-Sum clustering algorithm, we use the Gomory-Hu construction, and to characterize Single-Linkage, we use the Maximum Spanning Tree. Although at first glance it seems these properties are ‘obviously’ all that are necessary to characterize Max-Sum and Single-Linkage, we show that this is not the case, by investigating how subsets of properties interact. We conclude by proposing additions to the taxonomy of clustering paradigms currently in use.

Keywords: Clustering properties, axioms, submodularity, Q-Clustering

1. Introduction

Clustering is a ubiquitous task in unsupervised learning, finding application in a large array of fields other than Computer Science. Any task where one seeks to group together similar objects and separate dissimilar objects can be thought of as ‘Clustering’. Although the problem as defined leaves open a great deal of interpretation (Blum, 2009; Guyon et al., 2009), in this paper we concretely focus on a large class of objectives which can be optimized using Queyranne’s algorithm for submodular optimization (Queyranne, 1998; Rizzi, 2000).

This class of objectives for Q-Clustering was introduced and used in Narasimhan et al. (2006), which we analyze from an axiomatic perspective. Recently there has been a significant amount of work on clustering axioms and properties (Ackerman and Ben-David, 2008; Zadeh and Ben-David, 2009; Carlsson and Mémoli, 2010b), and we build in this direction to provide properties for the class of algorithms which can be expressed as optima of Queyranne’s optimization algorithm.

Narasimhan et al. (2006) considered 2 objectives to which we add and analyze a new objective. The two objectives in the original Q-clustering paper were Single-Linkage and Minimum Description Length (MDL). To these we add the Max-Sum objective, which seeks to maximize the sum of similarities inside clusters. Queyranne’s algorithm can optimize the Single-Linkage criterion perfectly for k clusters, but its perfect recovery is limited to only 2 clusters for Max-Sum and MDL. This is not surprising, since optimizing Max-Sum and MDL perfectly for $k > 2$ is NP-hard. However, there exist factor 2 approximation algorithms which proceed by cutting the $k - 1$ most expensive edges of the Gomory-Hu tree

(Gomory and Hu, 1961) associated with MDL and Max-Sum. Gomory-Hu Trees exist in general for any symmetric submodular function. Furthermore, the natural tree construction for Single-Linkage (SL) is the maximum spanning tree (MST), since the SL criterion can be optimized by cutting the $k - 1$ most expensive edges of the MST.

We focus on the underlying principles behind the Q-Clustering family of objectives, and arrive at the conclusion that they are uniquely characterized by cutting tree edges from well known tree constructions. Furthermore, we prove that they are the *unique* clustering functions using the mentioned tree constructions. These constructions are defined formally in section 1.2. We show that Max-Sum and Single-Linkage are axiomatically identical, except for which of the above trees they use in the course of their operation. Furthermore, we show that they are the *only* clustering functions determined by those trees.

The three objectives we consider are quite different. The Single-Linkage objective has the advantage that since we are only comparing similarities, the similarity measure can be an ordered set (arithmetic operations on the similarities need not be defined). We formalize the property that allows this criterion to only depend on the rank ordering of the similarities, so it is insensitive to monotone transformations of the similarities. This flexibility is useful in applications where only rankings are available, for example user studies in which humans only provide rankings instead of actual similarity scores. Unfortunately Single-Linkage is also very sensitive to outliers since all it takes to merge two clusters is a *single path* between the two clusters where all similarities are above a threshold. Since there are generally many paths between two clusters, this is a stringent requirement on the designers of the similarity matrix. The natural tree construction associated with the Single-Linkage Criterion is the Maximum Spanning Tree (MST) defined formally in the Section 1.2.

The second objective we consider is the Max-Sum objective, which doesn't suffer from the single-path problem of single-linkage, but has the tendency to create a single large cluster, since having more edges in the objective summation is usually better than fewer. The tree construction we use for this is the Gomory-Hu tree, explained formally in Section 1.2.

The third objective we consider is probabilistic in nature and is based on the Minimum Description Length principle. We are given a distribution for each data item, and we attempt to find clusters so that describing or encoding the clusters (separately) can be done using as few bits as possible. It is known that the problem of finding the optimal clusterings minimizing the description length is equivalent to the problem of minimizing a symmetric submodular function (Narasimhan et al., 2006). The tree construction associated with this objective is the cut-equivalent tree associated with the symmetric submodular MDL function. We do not provide a uniqueness theorem for MDL since it is not clear how to define some our axioms in this setting.

It is important to note that we make a stark contrast between *axioms* and *properties*. Although they both restrict the class of partitioning functions, we expect axioms to appeal to our intuition about clustering, whereas properties are simply restrictions on the class of partitioning functions, they need not appeal to any intuition about clustering.

1.1 Previous Work

In addition to submodular optimization algorithm by Queyranne (1998) generalized by Rizzi (2000), our formal framework is based on Kleinberg's (Kleinberg, 2003). We also adopt two

of the three axioms proposed in that paper, Consistency and Scale Invariance. We replace the Richness axiom of Kleinberg (2003) by its version for the case of fixed number of clusters, k -Richness.

An axiomatic characterization of Single-Linkage is available in Zadeh and Ben-David (2009), but they do not provide any characterization of Max-Sum or submodular objectives. Flake et al. (2004) present some algorithms which use Minimum Cut Trees for the purpose of Clustering, but they do not discuss axioms or uniqueness theorems. Another line of attack for characterizing clustering methods is using tools from Topology explored in Carlsson and Mémoli (2010a) and Carlsson and Mémoli (2008). Instead of fixing k , there are other ways of circumventing Kleinberg’s impossibility result analyzed in Ackerman and Ben-David (2009), and a characterization of the class of hierarchical clustering functions is given in Ackerman et al. (2010). Submodular objectives for clustering are explored in Jegelka and Bilmes (2010) and Narasimhan et al. (2006).

Finally, Balcan et al. (2008) and Awasthi and Zadeh (2010) present a framework which assumes that, given some data, a target clustering is achieved by interacting with a teacher. Interacting with a teacher is a departure from unsupervised learning.

1.2 Formal Preliminaries

A partitioning function acts on a set S of $n \geq 2$ points, and pairwise similarities among the points in S . The points in S are not assumed to belong to any specific set; the pairwise similarities are the only data the partitioning function has about them. Since we wish to deal with point sets that do not necessarily belong to a specific set, we identify the points with the set $S = \{1, 2, \dots, n\}$. We can then define a similarity function to be any function $s : S \times S \rightarrow \mathbb{R}^+$ such that for distinct $i, j \in S$, we have $s(i, j) > 0$, i.e. s must be positive symmetric, but there is no requirement of triangle inequality.

Sometimes we write $s = \langle e_1, e_2, \dots, e_{\binom{n}{2}} \rangle$ to mean the set of edges that exist between all pairs of n points. This list is *always* ordered by **decreasing** similarity. $w(e)$ is the weight of edge e which connects some two points i, j . So $w(e) = s(i, j)$.

A partitioning function is a function F that takes a similarity function s on $S \times S$ and returns a k -partitioning of S . A k -partitioning of S is a collection of k non-empty disjoint subsets of S whose union is S . The sets in $F(s)$ will be called its *clusters*. Two partitioning functions are equivalent if and only if they output the same partitioning on all values of s - i.e. functionally equivalent. In the next section we define several trees, each associated with a particular graph G .

A natural representation for a similarity function $s = \langle e_1, e_2, \dots, e_{\binom{n}{2}} \rangle$ is a complete weighted graph G_s , whose n nodes correspond to our objects, and edges correspond to similarity scores assigned by s . Note that when there is no ambiguity about which similarity function is being used, we drop the subscript s and simply use G . Also, when it is natural to think of the complete graph for the similarity function, we simply use s to refer to the graph, instead of G_s . Now some trees will be associated with a particular s .

1.2.1 MINIMUM CUT TREE

Let $G = (V, E, s)$ be any arbitrary weighted undirected connected graph with $|V| = n$. Two disjoint subsets A and B of V that also cover V define a cut in G . The sum of the weights

of the edges crossing the cut defines the cut value. For two nodes s, t , a minimum s - t cut is a cut of minimum value that separates nodes s and t . For an integer k , a minimum k -cut is a cut of minimum value that leaves exactly k connected components.

The Minimum Cut Tree of G - call it $\text{MCT}(G)$ - is a tree which has the same nodes as G , but a different set of edges and weights. The edges of the Minimum Cut Tree of G satisfy the following two properties.

- For every two nodes $s, t \in V$, the minimum cut in G that separates these points is given by cutting the smallest edge on the unique path between s and t in $\text{MCT}(G)$.
- For every two nodes $s, t \in V$, the weight of the smallest edge on the unique path (in $\text{MCT}(G)$) connecting them equals the size of a minimum s - t cut of G .

Note that $\text{MCT}(G)$ is *not* a subgraph of G as it has different edges and weight function. One may ask if such trees always exist, and indeed for every undirected graph, there always exists a min-cut tree, and they were initially introduced by Gomory and Hu (1961).

Minimum Cut Trees are not always unique, so we define a canonical MCT function which fixes a particular ordering on pairs of points (the lexicographical ordering), and uses the algorithm defined in (Gomory and Hu, 1961) and outlined momentarily. Under these conditions, the output of the Gomory-Hu algorithm will be deterministic (thus unique) and we will denote its value $\text{MCT}(G)$.

Given an input graph $G = (V, E)$ the Gomory-Hu algorithm maintains a partition of V , (S_1, S_2, \dots, S_t) and a spanning tree T on the vertex set $\{S_1, \dots, S_t\}$. Let w' be the function assigning weights to the edges of T . Tree T satisfies the following invariant. **Invariant:** For any edge (S_i, S_j) in T there are vertices a and b in S_i and S_j respectively, such that $w'(S_i, S_j) = f(a, b)$, where f denotes the minimum cut/maximum flow function, and the cut defined by edge (S_i, S_j) is a minimum a - b cut in G . This invariant is maintained for $n - 1$ steps, after which T is a Gomory-Hu tree of G (Vazirani, 2001). Each step involves a call to a standard s - t cut Min-Cut/Max-Flow algorithm.

1.2.2 SUBMODULAR FUNCTIONS

Fix a finite set S . Submodularity is a property enforced on functions mapping subsets of S set to the reals. Intuitively, a submodular function over the powerset demonstrates “diminishing returns”. From a practical perspective, there exist polynomial time algorithms for minimizing submodular functions. In this sense they are the discrete analog of convex functions (Lovász, 1983). Let S be a finite set. A function $f: 2^S \rightarrow \mathbb{R}$ is submodular iff for all subsets A and B of S we have,

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$$

Furthermore, a symmetric function is one for which $f(S \setminus A) = f(A)$. An important note is that Gomory-Hu trees work and can be generalized because the cut function is both submodular and symmetric. Any submodular symmetric function will induce a Gomory-Hu tree (Schrijver, 2003). An example of a symmetric submodular function is mutual information between two sets of random variables X and Y ,

$$I(X; Y) = \int_Y \int_X p(x, y) \log \left(\frac{p(x, y)}{p_1(x) p_2(y)} \right) dx dy,$$

where $p(x, y)$ is the joint probability distribution function of X and Y , and $p_1(x)$ and $p_2(y)$ are the marginal probability distribution functions of X and Y respectively. For a deeper discussion and proofs of submodularity, see Toshev (2010).

1.2.3 MAXIMUM SPANNING TREE

Given a connected, undirected weighted graph G , a spanning tree of G is a subgraph which is a tree and connects all the vertices. A single graph can have many different spanning trees. The weight of a spanning tree is computed as the sum of the weights of the edges in the spanning tree. A Maximum Spanning Tree (MST) is then a spanning tree with weight greater than or equal to every other spanning tree.

Similar to Minimum Cut Trees, MSTs also have a rich history. They can be computed efficiently by Kruskal’s algorithm. Maximum Spanning Trees are not always unique, but in the case that we fix an edge ordering, it is well known that they are unique for a particular G . We denote the canonical Maximum Spanning Tree of a graph as $\text{MST}(G)$.

1.2.4 SINGLE-LINKAGE

Single-Linkage is the clustering function which starts with all points in singleton clusters, and successively merges clusters until only k clusters are left. The similarity of two clusters is the similarity of the two most similar points inside differing clusters.

Another way to compute the Single-Linkage k -partitioning is to cut the $k - 1$ smallest edges of the Maximum Spanning Tree of G_s (Gower and Ross, 1969). It should be noted that the behavior of Single-Linkage is robust against small fluctuations in the weight of the edges in s , so long as the order of edges does not change, which can be readily seen from Kruskal’s algorithm.

1.2.5 MAX-SUM

The objective of Max-Sum is to maximize $\Lambda_s(\Gamma) = \sum_{c \in \Gamma} \sum_{i, j \in c} s(i, j)$ over all k -partitionings $\Gamma = \{A, B\}$. Finding the optimal partitioning is NP-hard for $k > 2$. However, for $k = 2$ finding the optimal Max-Sum 2-partitioning is the same as finding the global minimum cut and thus poly-time computable. Finding the overall minimum cut is equivalent to cutting the smallest edge of the Minimum Cut Tree, so for $k = 2$ Max-Sum can be reinterpreted as the algorithm which cuts the smallest edge of the Minimum Cut Tree.

Since it is not computationally feasible to optimize the above objective function, we define the following approximation algorithm which has a guaranteed approximation factor $2 - 2/k$ (Vazirani, 2001): simply iteratively find and remove the global minimum cut until exactly k connected components remain. This algorithm is called the “MaxCut” clustering function throughout.

2. Uniqueness Results

2.1 Axioms

Now in an effort to distinguish clustering functions from partitioning functions, we review some *axioms* (not properties) that one may like a clustering function to satisfy. Here is the

first one. If s is a similarity function, then define $\alpha \cdot s$ to be the same function with all similarities multiplied by α .

SCALE-INVARIANCE. *For any similarity function s , $1 \leq k \leq n$, and scalar $\alpha > 0$, we have*

$$F(s, k) = F(\alpha \cdot s, k)$$

This axiom simply requires the function to be immune to stretching or shrinking the data points linearly. It effectively disallows clustering functions to be sensitive to changes in units of measurement - which is desirable. We would like clustering functions to not have any predefined hard-coded similarity values in their decision process.

The next axiom ensures that the clustering function is “rich” and not crippled in types of partitioning it could output. For a fixed S , Let $\text{Range}(F(\bullet))$ be the set of all possible outputs while varying s .

k -RICHNESS. *Range($F(\bullet, k)$) is equal to the set of all k -partitionings of S*

In other words, if we are given a set of points such that all we know about the points are pairwise similarities, then for any partitioning Γ , there should exist a s such that $F(s) = \Gamma$. By varying similarities amongst points, we should be able to obtain all possible k -partitionings.

The next axiom is more subtle and was initially introduced in Kleinberg (2003), along with richness. We call a partitioning function “consistent” if it satisfies the following: when we increase similarities between points in the same cluster and decrease similarities between points in different clusters, we get the same result. Formally, we say that s' is a Γ -transformation of s if (a) for all $i, j \in S$ belonging to the same cluster of Γ , we have $s'(i, j) \geq s(i, j)$; and (b) for all $i, j \in S$ belonging to different clusters of Γ , we have $s'(i, j) \leq s(i, j)$. In other words, s' is a transformation of s such that points inside the same cluster are made more similar and points not inside the same cluster are made less similar.

CONSISTENCY. *Let s be a similarity function, and s' be a $F(s, k)$ -transformation of s .
 Then $F(s, k) = F(s', k)$*

In other words, suppose that we run the partitioning function F on s to get back a particular partitioning Γ . Now, with respect to Γ , if we shrink in-cluster similarities or expand between-cluster similarities and run F again, we should still get back the same result - namely Γ .

The difference between these and the axioms defined by Kleinberg (2003) is that at all times, the partitioning function F is forced to return a fixed number of clusters. If this were not the case, then the above axioms could never be satisfied by any function. In most popular clustering algorithms such as k -means, Single-Linkage, and spectral clustering, the number of clusters to be returned is determined beforehand – by the human user or other methods – and passed into the clustering function as a parameter. Note that it is not clear how to extend Consistency to the case where there is a distribution associated with every point, since there is not an obvious definition of similarity between two points, which is needed to define Consistency in terms of increasing or decreasing similarities.

Definition 1 *A Clustering Function is a partitioning function that satisfies Consistency, Scale Invariance, and k -Richness.*

We've already introduced two clustering functions.

Theorem 2 *Single-Linkage and Max-Sum are Clustering functions.*

Both Single-Linkage and Max-Sum are clustering functions, with proofs available by Zadeh and Ben-David (2009). So a natural question to ask is what types of *properties* (and not axioms) distinguish these two clustering functions from one another? We introduce two new properties that one may not expect all clustering functions to satisfy, but may at times be desirable.

2.2 Properties

For ease of notation, let $\text{MST}(s)$ be the Maximum Spanning Tree of G_s . Similarly, let $\text{MCT}(s)$ be the Minimum Cut Tree of G_s . Now we are ready to define two distinguishing properties.

MST-CONSISTENCY. *If s and s' are similarity functions such that $\text{MST}(s)$ and $\text{MST}(s')$ have the same minimum k -cut, then $F(s, k) = F(s', k)$*

In other words, a clustering function is MST-Consistent if it makes all its decisions based on the Maximum Spanning Tree. Note that this does not mean the algorithm must optimize any particular objective, just that its decisions are based on the MST. It is important to note that this property includes both the weights on the edges *and* the structure of the MST. Single-Linkage satisfies this property since it cuts the smallest edge $k - 1$ edges of the MST. Note that this is a *property*, not an axiom - we don't expect all clustering functions to make their decisions using the Maximum Spanning Tree. Later in this section we show that Single-Linkage is the *only* clustering function that is MST-Consistent.

Similarly define MCT-Consistency identical to MST-Consistency, with MST replaced with MCT.

MCT-CONSISTENCY. *If s and s' are similarity functions such that $\text{MCT}(s)$ and $\text{MCT}(s')$ have the same minimum k -cut, then $F(s, k) = F(s', k)$*

This property forces a clustering function to make all its decisions based on the Minimum Cut Tree. Max-Sum satisfies this property since it always cuts the smallest $k - 1$ edges of the minimum cut tree. We show that Max-Sum is the *only* clustering function that is MCT-Consistent.

Notice that both MST-Consistency and MCT-Consistency imply Scale-Invariance; meaning that if a function is either MST-Consistent or MCT-Consistent, then it is also Scale-Invariant. For this reason, whenever a function satisfies $\{\text{MCT}, \text{MST}\}$ -Consistency, we ignore Scale-Invariance.

2.3 Uniqueness Theorems

Shortly, we will be showing that Single-Linkage and Max-Sum are uniquely characterized by MST-Consistency and MCT-Consistency, respectively. Before doing this, we reflect on the relationships between subsets of our axioms and properties. In doing so, we show that

MST/MCT-Consistency properties are *not enough* to characterize SL and Max-Sum, as one might think from a shallow glance.

We will show that if any of the axioms or properties that we use for proving uniqueness were to be missing, then the uniqueness results do not hold. This means that all axioms and properties are really necessary to prove uniqueness of Single-Linkage and Max-Sum.

Theorem 3 *Consistency, MCT-Consistency, and k -Richness are necessary to characterize Max-Sum.*

Proof For each of the mentioned properties, we show that all the other properties and axioms together are not enough to uniquely characterize Max-Sum. To this end, for each of property, we exhibit an algorithm that acts differently than Max-Sum, and satisfies all the properties except for one. In other words, we show that without each of these properties, the remaining ones do not uniquely characterize Max-Sum.

Consistency is necessary. We define the Minimum Cut Tree Cuts family of partitioning functions. As usual, the task is to partition n points into k clusters. Let σ be a permutation function for the set of all k -partitionings of S . A particular member of the MCT cuts family computes the Max-Sum k -partitioning, then runs σ on the output of Max-Sum. The entire family is obtained by varying the particular permutation used. Since Max-Sum is k -Rich, and σ is a bijection, then all the members of MCT-Cuts are also k -Rich. Since Max-Sum is MCT-Consistent and σ does not look at the input, all the members of MCT-Cuts are also MCT-Consistent. However, it is not true that all members MCT-Cuts are Consistent. This is implied by theorem 6, which says the *only* Consistent member of MST-Cuts is Max-Sum itself, i.e. the case that σ is the identity permutation.

MCT-Consistency is necessary. Consider that Single-Linkage satisfies Consistency, Scale-Invariance, and k -Richness, but is obviously not the same function as Max-Sum. Thus MCT-Consistency is necessary.

k -Richness is necessary. Now consider the Constant clustering function which always returns the first $n - k + 1$ elements of S as a single cluster and returns the remaining k as singleton clusters (a singleton is a cluster with a single point in it), making a total of k clusters. Because this function does not look at s , it is trivially MST-Consistent, Consistent, and Scale-Invariant. However, it is not k -Rich because it always returns some singletons - i.e. we could never reach a k -partitioning that has no singletons. ■

Theorem 4 *Consistency, MST-Consistency, and k -Richness are necessary to characterize Single-Linkage.*

Proof omitted for space, but it very similar to proof of theorem 3. A summary of these results is available in table 1. Now that we have seen our properties do not trivially characterize neither Single-Linkage nor Max-Sum, we can move onto proving the uniqueness theorems.

Lemma 5 *Given a Consistent partitioning function F , and a similarity function s with edges in descending order of similarity*

$$s = \langle e_1, e_2, \dots, e_p, e_q, \dots, e_{\binom{n}{2}} \rangle$$

	Consistency	k -Richness	MST-Consistency	MCT-Consistency
Single-Linkage	✓	✓	✓	×
Max-Sum	✓	✓	×	✓
MST cuts family	×	✓	✓	×
MCT cuts family	×	✓	×	✓
Constant partitioning	✓	×	✓	✓

Table 1: Overview of discussed partitioning functions. Even if one were to consider more partitioning functions, as a consequence of theorems 7 and 6, the Single-Linkage and Max-Sum rows are unique amongst all partitioning functions.

then for all $k > 0$, if e_p and e_q are both inner edges or both outer edges (w.r.t. $F(s, k)$), we have

$$F(\langle e_1, e_2, \dots, e_q, e_p, \dots, e_{\binom{n}{2}} \rangle, k) = F(s, k)$$

Proof In other words, whenever we have two edges of the same type (inner or outer), in neighboring positions in the edge ordering of s , we can swap their positions while maintaining the output of F . This is true because if both e_p and e_q are outer edges, then we can shrink e_p until $w(e_p) < w(e_q)$ all the while preserving the output of F (by Consistency). Similarly, if both e_p and e_q are inner edges, we can expand e_q until $w(e_p) < w(e_q)$. ■

Theorem 6 *Max-Sum is the only MCT-Consistent Clustering function.*

Proof Let F be any Consistent, k -Rich, MCT-Consistent clustering function, and let s be any similarity function on n points. k is an integer with $1 \leq k \leq n$. We want to show that for all s, k , $F(s, k) = \text{MS}(s, k)$, where $\text{MS}(s, k)$ is the result of Max-Sum on s . For this purpose, we introduce the partitioning Γ as whatever the output of MS is on s , so $\text{MS}(s, k) = \Gamma$. Whenever we say “inner” or “outer” edge for this proof, we mean with respect to Γ .

By k -Richness of F , there exists an s_1 such that $F(s_1, k) = \text{MS}(s, k) = \Gamma$. Now, through a series of transformations that preserve the output of F , we transform s_1 into s_2 , then s_2 into s_3 , ..., until we arrive at s .

1. By k -Richness, we know there exists an s_1 such that $F(s_1, k) = \text{MS}(s, k) = \Gamma$.
2. Using scale-invariance, we can linearly shrink all edges in s_1 until they are all smaller than the smallest edge in s_1 , call the result s_2 .
3. Now using consistency, we can expand all inner edges of s_1 until they are exactly of the same weight as they appear in s . Call the result s_3 . Thus, s_3 has all inner edges set to exactly the same weight as in s , but the outer edges of s_3 are all smaller than their counterparts in s .
4. Using consistency, we can shrink all outer edges of s_3 until the sum of all outer edges of s_3 is smaller than the smallest inner edge of s_3 (and s). Call the result s_4 .

5. By lemma 5 we can reorder all outer edges in s_4 until their order among themselves is the same order as they appear in s . Call the result s_5 .
6. For this step it helps to review the construction of Gomory-Hu trees outlined in section 1.2.1. Consider that the sum of all outer edges in s_5 is less than any single inner edge in s_5 . Thus cutting all outer edges in s_5 is cheaper than cutting any single inner edge. Furthermore, the removal of these outer edges in G_{s_5} results in k connected components. To construct a Gomory-Hu tree for G_{s_5} , we build the tree by maintaining the Gomory-Hu invariant while querying points in differing clusters of Γ with each step. The standard s - t MinCut algorithm in the Gomory-Hu iteration is guaranteed to return a subset of the outer edges of s_5 , since the sum of all outer edges has less weight than any individual inner edge. So after the first $k - 1$ iterations of Gomory-Hu, the intermediate tree T will have $k - 1$ edges and k supernodes, and the weight of the edges of T will be smaller than any inner edge of s . We then run Gomory-Hu to completion to obtain the $T_{s_5} = \text{MCT}(s_5)$. We do the same for s , and call the result T_s .
7. Since T_s and T_{s_5} are trees, their minimum k -cut is given by cutting their $k - 1$ smallest edges respectively. However, since both of their $k - 1$ lightest edges correspond to cutting the outer edges of s_5 , then T_s and T_{s_5} have the same minimum k -cut. Thus by MCT-Consistency we can transform s_5 to s while maintaining the output of F .
8. Thus we have $F(s_5, k) = F(s, k) = \Gamma$.

We started with any s , and showed that $F(s, k) = \Gamma = \text{MS}(s, k)$. We also know that Max-Sum satisfies all 3 axioms and MCT-Consistency. Thus it is uniquely characterized. \blacksquare

Theorem 7 *Single-Linkage is the only MST-Consistent Clustering function.*

Proof

Let F be any Consistent, k -Rich, MST-Consistent clustering function, and let s be any similarity function on n points. k is an integer with $1 \leq k \leq n$. We want to show that for all s, k , $F(s, k) = \text{SL}(s, k)$, where $\text{SL}(s, k)$ is the result of Single-Linkage on s . For this purpose, we introduce the partitioning Γ as whatever the output of SL is on s , so $\text{SL}(s, k) = \Gamma$. Whenever we say “inner” or “outer” edge for this proof, we mean with respect to Γ .

By k -Richness of F , there exists an s_1 such that $F(s_1, k) = \text{SL}(s, k) = \Gamma$. Now, through a series of transformations that preserve the output of F , we transform s_1 into s_2 , then s_2 into s_3, \dots , until we arrive at s .

1. By k -Richness, we know there exists an s_1 such that $F(s_1, k) = \text{SL}(s, k) = \Gamma$.
2. Using scale-invariance, we can linearly shrink all edges in s_1 until they are all smaller than the smallest edge in s_1 , call the result s_2 .
3. Now using consistency, we can expand all inner edges of s_1 until they are exactly of the same weight as they appear in s . Call the result s_3 . Thus, s_3 has all inner edges

set to exactly the same weight as in s , but the outer edges of s_3 are all smaller than their counterparts in s .

4. Using consistency, we can shrink all outer edges of s_3 until the sum of all outer edges of s_3 is smaller than the smallest inner edge of s_3 (and s). Call the result s_4 .
5. By lemma 5 we can reorder all outer edges in s_4 until their order among themselves is the same order they appear in s , we can do the same for the inner edges. Call the result s_5 .
6. By Kruskal’s algorithm for MST, s and s_5 have the same MST, call them T_s and T_{s_5} . Since T_s and T_{s_5} are trees, their minimum k -cut is given by cutting their $k - 1$ smallest edges respectively. However, since both of their $k - 1$ lightest edges correspond to cutting the outer edges of s_5 , then T_s and T_{s_5} have the same minimum k -cut. Thus by MST-Consistency we can transform s_5 to s while maintaining the output of F .
7. Thus we have $F(s_5, k) = F(s, k) = \Gamma$.

We started with any s , and showed that $F(s, k) = \Gamma = \text{SL}(s, k)$. We also know that Single-Linkage satisfies all 3 axioms (Zadeh and Ben-David, 2009) and MST-Consistency. Thus it is uniquely characterized. ■

3. Conclusions & Future directions

In this paper we have characterized two clustering algorithms which are usually treated separately with differing motivating principles. Using our framework, one can aim to build a suite of abstract properties of clustering that will induce a taxonomy of clustering paradigms. Such a taxonomy should serve to help utilize prior domain knowledge to allow educated choice of a clustering method that is appropriate for a given clustering task.

The chief contributions of this paper are the characterizations of clustering through associated tree constructions. The tree construction relevant for Max-Sum turned out to be the Gomory-Hu tree, and the tree construction for Single-Linkage was the Maximum Spanning Tree. It is expected that the general Gomory-Hu tree construction for *any* submodular function can also be used for characterizing other objectives, but it is not immediately clear how to change axioms such as Consistency to achieve this and we leave it for future work.

Our contribution provides insight into the connection between Submodularity, Single-Linkage, and Max-Sum clustering functions. For the latter two, we show they are axiomatically identical except for one property. By considering the listing in table 1, we demonstrate the type of desired taxonomy of clustering functions based on the properties each satisfies.

Although at first glance it seems these properties are ‘obviously’ all that are necessary to characterize Max-Sum and Single-Linkage, we show that this is not the case, by way of Theorems 3 and 4. To investigate the ramifications of algorithms which satisfy only a subset of our properties, we introduced the MCT Cuts family of partitioning functions, of which Max-Sum is the only Consistent member. The uniqueness theorems came about as a result of forcing functions to focus on Minimum/Maximum Cut/Spanning Trees.

Acknowledgments

We would like to acknowledge Shai Ben-David, Stefanie Jegelka, Avrim Blum, and Ashish Goel for very valuable discussions.

References

- Margarita Ackerman and Shai Ben-David. Measures of Clustering Quality: A Working Set of Axioms for Clustering. In *Advances in Neural Information Processing Systems: Proceedings of the 2008 Conference*, 2008.
- Margarita Ackerman and Shai Ben-David. Clusterability: A theoretical study. *Proceedings of AISTATS-09, JMLR: W&CP*, 5:1–8, 2009.
- Margarita Ackerman, Shai Ben-David, and David Loker. Characterization of Linkage-based Clustering. *COLT 2010*, 2010.
- Pranjal Awasthi and Reza Bosagh Zadeh. Supervised Clustering. *Neural Information Processing Systems (NIPS 2010)*, 2010.
- M.-F. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008.
- Avrim. Blum. Thoughts on clustering. In *NIPS Workshop on Clustering Theory*, 2009.
- Gunnar Carlsson and Facundo Mémoli. Persistent Clustering and a Theorem of J. Kleinberg. *ArXiv e-prints*, 2008.
- Gunnar Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *The Journal of Machine Learning Research*, 99:1425–1470, 2010a. ISSN 1532-4435.
- Gunnar Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *The Journal of Machine Learning Research*, 99:1425–1470, 2010b.
- G.W. Flake, R.E. Tarjan, and K. Tsioutsouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1(4):385–408, 2004.
- RE Gomory and TC Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, pages 551–570, 1961.
- J.C. Gower and GJS Ross. Minimum spanning trees and single linkage cluster analysis. *Applied Statistics*, 18(1):54–64, 1969.
- Isabelle Guyon, Ulrike Von Luxburg, and Rob .C. Williamson. Clustering: Science or art. In *NIPS 2009 Workshop on Clustering Theory*, 2009.
- Stefanie Jegelka and Jeff Bilmes. Cooperative cuts: Graph cuts with submodular edge weights. Technical report, Technical Report, 2010.
- Jon Kleinberg. An Impossibility Theorem for Clustering. In *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*. MIT Press, 2003.

- Laszlo Lovász. Submodular functions and convexity. *Mathematical programming: the state of the art*, pages 235–257, 1983.
- Mukund Narasimhan, Nebojsa Jojic, and Jeff Bilmes. Q-clustering. *Advances in Neural Information Processing Systems*, 18:979, 2006.
- Maurice Queyranne. Minimizing symmetric submodular functions. *Mathematical Programming*, 82(1):3–12, 1998.
- Romeo Rizzi. Note on minimizing symmetric set functions. *Combinatorica*, 20(3):445–450, 2000.
- A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*. Springer Verlag, 2003. ISBN 3540443894.
- A Toshev. Submodular function minimization. *University of Pennsylvania, Philadelphia, Tech. Rep*, 2010.
- Vijay Vazirani. *Approximation algorithms*. Springer Verlag, 2001. ISBN 3540653678.
- Reza Bosagh Zadeh and Shai Ben-David. A Uniqueness Theorem for Clustering. In *Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, 2009.